Scientific Research Publishing

Journal of Mathematical Finance, 2020, 10, 1-9
https://www.scirp.org/journal/jmf
ISSN Online: 2162-2442
ISSN Print: 2162-2434

A Clustering Method to Solve Backward Stochastic Differential Equations with Jumps

Liangliang Zhang

NatWest Markets, Stamford, CT, USA
Email: liangliangzhang81@qq.com

How to cite this paper: Zhang, L.L. (2020) A Clustering Method to Solve Backward Stochastic Differential Equations with Jumps. *Journal of Mathematical Finance*, **10**, 1-9.
https://doi.org/10.4236/jmf.2020.101001

Received: October 8, 2019
Accepted: December 10, 2019
Published: December 13, 2019

Abstract

In this paper, we introduce a clustering method to approximate the solution to a general Backward Stochastic Differential Equation with Jumps (BSDEJ). We show the convergence of the sequence of approximate solutions to the true one. The method is implemented for an application in finance. Numerical results show that the method is efficient.

Keywords

Backward Stochastic Differential Equation with Jumps, Jump Diffusion, Clustering, Weak Convergence

1. Introduction

Solving BSDEs is of great importance in mathematical finance. The derivatives pricing problems, dynamic portfolio choice problems and even dynamic stochastic general equilibrium problems can be related to various types of BSDEs. However, due to its recursive nature, a BSDE with jumps cannot be directly solved via Monte Carlo method.

In this paper, we apply the methodology to evaluate conditional expectations, which is originally introduced in [1], to solve BSDEs with jumps. This novel method simplifies the structures of the approximate solution to the true one, by running a local approximation. Under the assumption that the solution is continuous, we can show that a linear regression approximation locally will be accurate enough with very limited computation budget and convergence is shown. To be specific, we use Monte-Carlo simulation and machine learning clustering method to divide the state space into small clusters, in which we approximate the true solutions via linear regression.

Recent literature of machine learning method to solve PDEs/BSDEs can be found in, for example, [2] [3] [4] [5] and [6]. In those references, the authors perform a brute-force machine learning approximation to the true solutions and the methods are therefore time-consuming. Our method can serve as a generalization of the above methods, in that, when the number of clusters is set to be 1, our method degenerates to the aforementioned ones.

The organization of this paper is as follows. Section 2 describes the method and convergence proof. Section 3 discusses an application in finance and Section 4 concludes. All the proofs can be found in Appendix.

2. A Formal Description of Our Approach

2.1. The BSDE

Throughout the paper, we consider a filtered probability space $\left(\Omega,\mathcal{F},(\mathcal{F}_t)_{0\le t\le T},\mathbb{P}\right)$, with $T\in R^+$. The space is supporting a *d*-dimensional Brownian motion $W_t=\left(W_t^1,\cdots,W_t^d\right)$ and a Poisson random measure *N* on $\mathcal{B}([0,T])\otimes\mathcal{E}$, where $\mathcal{B}([0,T])$ is the Borel σ-algebra on $[0,T]$ and $(E,\mathcal{E})$ is a measurable space. Define $E:=\mathbb{R}^q$ and $\mathcal{E}$ as the Borel σ-algebra on *E*. $\mathbb{P}$ is the probability measure on $\mathcal{F}$. The filtration $(\mathcal{F}_t)_{0\le t\le T}$ is completed with all $\mathbb{P}$-null sets, right-continuous and $\mathcal{F}_t=\mathcal{F}_t^{W,N}$ is generated by $\left(W_t,N(\cdot,[0,t],\cdot)\right)$ for $t\in[0,T]$. Assume that $\mathcal{F}=\mathcal{F}_T^{W,N}$ and *W* and *N* are mutually independent under $\mathbb{P}$. Suppose that the compensating measure of *N* is $\nu(\mathrm{d}t,\mathrm{d}e):=\nu(\mathrm{d}e)\mathrm{d}t$, where ν is a σ-finite measure on $(E,\mathcal{E})$ satisfying $\int_E\left(1\wedge|e|^2\right)\nu(\mathrm{d}e)<\infty$. The corresponding compensated Poisson random measure is defined by $\tilde{N}(\omega,\mathrm{d}t,\mathrm{d}e):=N(\omega,\mathrm{d}t,\mathrm{d}e)-\nu(\mathrm{d}e)\mathrm{d}t$.

The BSDEJ under consideration, whose solution is denoted by (X,Y,Z,U), is

$$\begin{aligned}&\mathrm{d}X_t=\mu(t,X_t)\mathrm{d}t+\sigma(t,X_t)\mathrm{d}W_t+\int_E\gamma(t-,X_{t-},e)\tilde{N}(\mathrm{d}t,\mathrm{d}e)\qquad X_0=x_0\\&\mathrm{d}Y_t=-f(t,X_t,Y_t,Z_t,V_t)\mathrm{d}t+Z_t\mathrm{d}W_t+\int_E U_{t-}(e)\tilde{N}(\mathrm{d}t,\mathrm{d}e)\qquad Y_T=\phi(X_T)\end{aligned}\tag{1}$$

where $V_t=\int_E U_t(e)\rho(e)\nu(\mathrm{d}e)$ for a given smooth and bounded function ρ. The vector of state variables $X_t\in\mathcal{F}_t$ is *r*-dimensional and satisfies the Forward-SDE (FSDE) indicated. The standard Brownian motion *W* is *d*-dimensional, $Y_t\in\mathcal{F}_t$ is 1-dimensional, $Z_t\in\mathcal{F}_t$ is *d*-dimensional, $U_t(e)\in\mathcal{F}_t$ is *q*-dimensional, $V_t\in\mathcal{F}_t$ is 1-dimensional and $\tilde{N}$ is *q*-dimensional. (X,Y,Z,U) is called the solution to the BSDEJ system (1). We assume that $r\le d$ throughout the paper without loss of generality.

2.2. The Numerical Approximation Scheme

This section describes the construction of the approximate solution to the BSDEJ (1). The intermediate solutions can be obtained by the following equations, according to [7]

$$\begin{aligned}&Y_t=u(t,X_t)=\mathbb{E}\left[\phi(X_T)+\int_t^T f(v,X_v,Y_v,Z_v,V_v)\mathrm{d}v\mid\mathcal{F}_t\right]\\&Z_t=\partial_x u(t,X_t)\sigma(t,X_t)\\&U_t(e)=u(t,X_t+\gamma(t,X_t,e))-u(t,X_t)\end{aligned}\tag{2}$$

where (Y_t, Z_t, U_t) are functions of (t, X_t). It is therefore crucial to be able to evaluate the above conditional expectations for each *t*.

Suppose that the domain of state variables X is $\mathbb{D} \subset \mathbb{R}^r$ [1]. Partition $\mathbb{D}$ by a class of disjoint sets $\{U^k\}_{k=1}^K$. Denote the distance $d_U = \sup_{x,y\in U} |x-y|$, where $|\cdot|$ is the canonical Euclidean distance. It is known that the evaluation of conditional expectation problems are essentially regression ones, in the following sense

$$\mathbb{E}_t\left[\psi(X_T)\right] \equiv \arg\min_{\phi\in\Phi(\mathbb{D})} \mathbb{E}\left[\left|\psi(X_T) - \phi(t, X_t)\right|^2\right] \tag{3}$$

where $\Phi(\mathbb{D})$ is an appropriate function space with domain $\mathbb{D}$ [2]. First, we need the following result.

Lemma 1. *We have*

$$\mathbb{E}_t\left[\psi(X_T)\right] \equiv \lim_{i\to\infty} \arg\min_{\phi\in\Phi(\mathbb{D}_i)} \mathbb{E}\left[\left|\psi(X_T) - \phi(t, X_t)\right|^2 \mathbf{1}_{X_t\in\mathbb{D}_i}\right] \tag{4}$$

in L^2 sense.

With the above result in mind, we will always write $\mathbb{D}_i$ as $\mathbb{D}$ and assume that $\mathbb{D}$ is bounded and closed. Many methods have been emerging to find global solutions to the above Equation (3). However, in this paper, we follow a divide-and-conquer approach, by trying to fit the regression locally in each of the subspace U^k, for $k = 1, 2, \cdots, K$. This means, we should try to evaluate

$$\phi_k(t, X_t) := \arg\min_{\phi_k\in\Phi(U^k)} \mathbb{E}\left[\left|\psi(X_T)\mathbf{1}_{X_t\in U^k} - \phi(t, X_t)\mathbf{1}_{X_t\in U^k}\right|^2\right] \tag{5}$$

where $\Phi(U^k)$ is an appropriate function space with domain U^k, under the constraint $\sum_{k=1}^K \phi_k(t,x)\mathbf{1}_{x\in U^k} \in \Phi$. ([1], Theorem 23) concludes that, indeed, we have $\phi(t,x) \equiv \sum_{k=1}^K \phi_k(t,x)\mathbf{1}_{x\in U^k}$. In ([1], Theorem 24), the condition $\sum_{k=1}^K \phi_k(t,x)\mathbf{1}_{x\in U^k} \in \Phi$ is relaxed under certain conditions. In this paper, we will use the two aforementioned theorems and show that polynomial regressions and Taylor expansions for ϕ_k and ψ, applying a Monte Carlo simulation to generate samples and use machine learning clustering to partition the simulated paths, will result in convergent numerical methods. In order to use Taylor expansion, we need ψ to be C^l-smooth, meaning it should have bounded derivatives up to order l in $\mathbb{D}$. This requires us to approximate it using C^l-smooth functions. The following assumption applies throughout the paper.

Assumption 1. (On ψ). *Assume that, in some sense, ψ can be approximated by a sequence of C^l-smooth functions $\{\psi_i\}$, with l finite or ∞. Moreover, we have*

[1] Of course, set $\mathbb{D}$ can be time-dependent and unbounded. For brevity we ignore the time variable throughout the paper. But we will use a sequence of bounded and closed subsets of $\mathbb{D}$, denoted by $\{\mathbb{D}_i\}_{i=1}^\infty$ and assume that $\mathbb{D}_i \subset \mathbb{D}_{i+1}$ and $\bigcup_{i=1}^\infty \mathbb{D}_i = \mathbb{D}$. Then, we will choose a sufficiently large *I* and perform computations in $\mathbb{D}_i$ instead of $\mathbb{D}$.

[2] For example, we can ask $\Phi(\mathbb{D}) := \{\phi \mid \phi(t, X_t) \text{ is } L^2 \text{ for all } t\in[0,T]\}$.

$$\lim_{i\to\infty}\mathbb{E}_t\left[\left|\psi\left(X_t\right)-\psi_i\left(X_t\right)\right|^2\right]=0 \tag{6}$$

for any $t\in[0,T]$.

Remark 2. *For example,* ψ_i *can converge to* ψ *in a point-wise sense or* L^2 *sense. Because of Assumption* 1, *we will always assume that* ψ *is* C^l *-smooth.*

Then, we have the following Theorems.

Theorem 3. (On Polynomial Regression). *Assume that*[3]

$$\hat{\phi}_{k,J}\left(t,X_t\right):=\arg\min_{\mathcal{P}_J\in\mathcal{P}_J\left(U^k\right)}\mathbb{E}\left[\left|\psi\left(X_T\right)\mathbf{1}_{X_t\in U^k}-\mathcal{P}_J\left(X_t\right)\mathbf{1}_{X_t\in U^k}\right|^2\right] \tag{7}$$

where $\mathcal{P}_J\left(U^k\right)$ is the space of all polynomials of degree equal to or less than J with domain U^k. Then, we have

$$\phi\left(t,X_t\right)=\lim_{\max\limits_{1\le k\le K} d_{U^k}\to 0+}\sum_{k=1}^{K}\hat{\phi}_{k,J}\left(t,X_t\right)\mathbf{1}_{X_t\in U^k} \tag{8}$$

in L^2 sense with $J\ge 1$ fixed.

Theorem 4 (On Taylor Expansion). *Assume that*

$$\psi\left(x\right)\cong\hat{\psi}_J\left(x,x_t^k\right)=\sum\nolimits_{|\alpha|\le J}\frac{\partial_\alpha\psi\left(x_t^k\right)}{\alpha!}\left(x-x_t^k\right)^\alpha \quad \textit{and}$$

$$\hat{\phi}_{k,J}\left(t,X_t^k\right)=\mathbb{E}_t\left[\hat{\psi}_J\left(X_T,X_t^k\right)\mathbf{1}_{X_t^k\in U^k}\right] \tag{9}$$

where $X_t^k\in U^k$. Then, we have

$$\phi\left(t,X_t\right)=\lim_{\max\limits_{1\le k\le K} d_{U^k}\to 0+}\sum_{k=1}^{K}\hat{\phi}_{k,J}\left(t,X_t\right)\mathbf{1}_{X_t\in U^k} \tag{10}$$

in L^2 sense with $J\ge 1$ fixed.

The following theorem establishes the convergence under Monte Carlo simulation.

Theorem 5 (On Monte Carlo Simulation). *Assume that there is a time-discretization* $\left\{t_j\right\}_{j=1}^n$ *and we obtain M samples* $\left\{X_{t_j}^m\right\}_{j,m=1,1}^{n,M}$. *Then, we have*

$$\phi\left(t,X_t\right)=\lim_{M\to\infty}\lim_{\max\limits_{1\le k\le K} d_{U^k}\to 0+}\frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K}\hat{\phi}_{k,J}\left(t,X_t^m\right)\mathbf{1}_{X_t^m\in U^k} \tag{11}$$

where $\hat{\phi}_{k,J}$ is obtained via OLS in the cases of polynomial regression and Taylor expansion, in L^2 sense with $J\ge 1$ fixed.

Theorem 5 is ultimately the convergence result for the methodology we propose. We first generate *M* copies of the sample paths of *X* from 0 to *T*. Then, we use MiniBatchKMeans function in Python programming language to partition the *M* copies of simulated values of *X*, at each time $t\in[0,T]$, into *K* clusters. With each cluster U^k for $k=1,2,\cdots,K$, we perform the analysis as shown above and concatenate different pieces together to formulate the values for the conditional expectations at time *t*. The methodology works backwards in time.

[3] The equation below means that we run a lead-lag polynomial regression within each U^k, with *Y* variable $\psi\left(X_T\right)$ and *X* variable X_t.

Of course, there are many different choices of $\hat{\phi}_{k,J}$, depending on how we want to approximate the conditional expectations within U^k.

With the above discussions on the evaluation of conditional expectations, we can start to describe the main numerical method for a BSDEJ. First, define a set of time nodes $\pi_n = \{t_i\}_{i=1}^n$, with $t_0 = 0$ and $t_n = T$. Suppose that $t_i - t_{i-1} \equiv h = \frac{T}{n}$. Second, we simulate M copies of the forward process X at the discretization nodes $\{t_i\}_{i=1}^n$. Third, we iterate backwards in time using the methodologies to evaluate conditional expectations outlined above. At time interval $[t_{n-1}, t_n)$, use MiniBatchKMeans method to cluster $\{X_{t_{n-1}}^{\pi_n,m}\}_{m=1}^M$ into K disjoint subsets $\{U_{t_{n-1}}^k\}_{k=1}^K$. Within each cluster $U_{t_{n-1}}^k$, apply polynomial regression or Taylor expansion method to evaluate $\hat{\phi}_{\pi_n,M,k,J}(t_{n-1}, X_{t_{n-1}}^{\pi_n,m})$, where we add subscript π_n to remind the readers that this intermediate function depends on time discretization now and M to denote Monte Carlo simulation dependence. After we obtain $\{\hat{\phi}_{\pi_n,M,k,J}\}_{k=1}^K$, we can treat it as the intermediate solution to the BSDEJ at time t_{n-1}. In order to evaluate Z and V, taking Z as an example, we use the relation $Z_t \cong \frac{1}{h}\mathbb{E}_t[\Delta Y_t \Delta W_t]$. The equations for Z and V are stated in ([7], Equation 2.10). When we obtain intermediate approximate solution $(Y_{t_{n-1}}^{\pi_n,M,k,J}, Z_{t_{n-1}}^{\pi_n,M,k,J}, V_{t_{n-1}}^{\pi_n,M,k,J})$, we plug the tuple into the driver f of the BSDEJ and $Y_{t_{n-1}}^{\pi_n,M,k,J} + hf(t_{n-1}, X_{t_{n-1}}^{\pi_n}, Y_{t_{n-1}}^{\pi_n,M,k,J}, Z_{t_{n-1}}^{\pi_n,M,k,J}, V_{t_{n-1}}^{\pi_n,M,k,J})$ is the terminal condition for the evaluations in time interval $[t_{n-2}, t_{n-1})$. Iterate until we reach t_0.

2.3. Convergence Results

First, we introduce necessary spaces to facilitate the discussions of convergence results. For an $\mathbb{R}^r$-valued function $x:[0,T] \to \mathbb{R}^r$, let the sup-norm be

$$\|x\|_{[a,b]} := \sup\{|x_t|, t \in [a,b]\}. \tag{12}$$

We shall use the following spaces for stochastic processes with $p \geq 2$

- $\mathbb{S}_r^p[s,t]$ is the set of $\mathbb{R}^r$-valued adapted càdlàg processes X such that

$$\|X\|_{\mathbb{S}_r^p[s,t]} := \mathbb{E}\left[\mathrm{e}^{\theta(t-s)}\|X(\omega)\|_{[s,t]}^p\right]^{\frac{1}{p}} < \infty. \tag{13}$$

Here $\theta \geq 0$ is a nonnegative constant. We sometimes write $\mathbb{S}_r^p[s,t]$ as $\mathbb{S}^p$ if doing so causes no ambiguity. The same is true for the spaces to be defined below.

- $\mathbb{H}^p[s,t]$ is the set of progressively measurable $\mathbb{R}^d$-valued processes Z such that

$$\|Z\|_{\mathbb{H}^p[s,t]} := \mathbb{E}\left[\left(\int_s^t \mathrm{e}^{\theta(v-s)}|Z_v|^2\,\mathrm{d}v\right)^{\frac{p}{2}}\right]^{\frac{1}{p}} < \infty. \tag{14}$$

- $\mathbb{J}^p[s,t]$ is the set of q-dimensional processes $\psi = \{\psi^i, 1 \leq i \leq q\}$, $\psi^i : \Omega \times [0,T] \times E \to \mathbb{R}$ which is $\mathbb{F} \times \mathcal{B}([0,T]) \times \mathcal{B}(E)$-measurable and satisfy

$$\|\psi\|_{\mathbb{J}^p[s,t]} := \mathbb{E}\left[\left(\sum_{j=1}^{q}\int_s^t\int_E \mathrm{e}^{\theta(v-s)}\left|\psi^i(\omega,v,e)\right|^2 v^i(\mathrm{d}e)\,\mathrm{d}v\right)^{\frac{p}{2}}\right]^{\frac{1}{p}} < \infty. \tag{15}$$

- $\mathcal{K}^p[s,t]$ is the set of processes (Y,Z,ψ) in the space $\mathbb{S}^p[s,t]\times\mathbb{H}^p[s,t]\times\mathbb{J}^p[s,t]$ with the norm

$$\|(Y,Z,\psi)\|_{\mathcal{K}^p[s,t]} := \left(\|Y\|_{\mathbb{S}^p[s,t]}^p + \|Z\|_{\mathbb{H}^p[s,t]}^p + \|\psi\|_{\mathbb{J}^p[s,t]}^p\right)^{\frac{1}{p}}. \tag{16}$$

Let $C_b^g(D)$ be the space of bounded functions that have continuous and bounded derivatives up to order g in the domain $D\subset\mathbb{R}^r$ and $C^g(D)$ the space of functions that have continuous derivatives up to order g. In what follows, we will only consider the norms with $\theta>0$ and $p=2$. In order to provide the final convergence result, we need the following assumption.

Assumption 6 (On BSDEJ). *Assume that the BSDEJ* (1) *admits a unique solution* $S_t=(Y_t,Z_t,V_t)$ *in* $\mathcal{K}^p[0,T]$ *and the time discretized solution* $S_t^{\pi_n} := (Y_t^{\pi_n},Z_t^{\pi_n},V_t^{\pi_n})$ *converges to* $S_t=(Y_t,Z_t,V_t)$ *in* $\mathcal{K}^p[0,T]$ *sense.*

Then, the following convergence result is valid to justify the methodology we introduce above.

Theorem 7 (Main Convergence Result). *Under Assumptions* 1 *and* 6, *we have*

$$\lim_{h\to 0+}\lim_{M\to\infty}\lim_{\max\limits_{1\le k\le K} d_{U^k}\to 0+}\left\|S^{\pi_n,M,K,J}-S\right\|_{\mathcal{K}^p[0,T]} = 0 \tag{17}$$

where $S_t^{\pi_n,M,K,J} = \sum_{k=1}^{K} S_t^{\pi_n,M,k,J}\mathbf{1}_{X_t\in U_t^k}$ and $S_t^{\pi_n,M,k,J} = \left(Y_t^{\pi_n,M,k,J}, Z_t^{\pi_n,M,k,J}, V_t^{\pi_n,M,k,J}\right)$.

3. Applications in Finance

The BSDE associated with American option prices can be found in [8]. We numerically solve ([8], Equation 2.10) using the methodology introduced in Section 2. Performance is documented below. Here M is the number of sample paths and H is the number of time discretization nodes. The time to maturity of the American call option is $T=0.50$. S_0 is the initial price of the underlying asset and the strike price is 100. Table 1 and Table 2 contain efficiency results.

4. Conclusion

In this paper, we propose a new machine learning method, based on clustering, to run [9] type regression to solve BSDEs with jumps. Convergence proof is given and numerical results show good performance. The generalization of the proposed methodologies to more complicated BSDEs with jumps, for example, the Mckean-Vlasov type BSDEs, is of interest. We leave it for future research.

Table 1. Numerical results when $M = 50000$.

$M = 50000$ $N = 50$	S_0	Benchmark	Clustering	Ave. Time (s)	Rel. Error
$\sigma = 0.2$	80	0.219	0.222	12.56	1.37%
	90	1.386	1.412	12.56	1.88%
	100	4.783	4.790	12.56	0.15%
	110	11.098	11.131	12.56	0.30%
	120	20.000	19.984	12.56	−0.08%
$\sigma = 0.4$	80	2.689	2.736	12.56	1.75%
	90	5.722	5.778	12.56	0.98%
	100	10.239	10.274	12.56	0.34%
	110	16.181	16.204	12.56	0.14%
	120	23.360	23.413	12.56	0.23%

Table 2. Numerical results when $M = 75000$.

$M = 75000$ $N = 100$	S_0	Benchmark	Clustering	Ave. Time (s)	Rel. Error
$\sigma = 0.2$	80	0.219	0.223	42.716	1.83%
	90	1.386	1.397	42.716	0.79%
	100	4.783	4.814	42.716	0.65%
	110	11.098	11.114	42.716	0.14%
	120	20.000	19.979	42.716	−0.11%
$\sigma = 0.4$	80	2.689	2.699	42.716	0.37%
	90	5.722	5.749	42.716	0.47%
	100	10.239	10.299	42.716	0.59%
	110	16.181	16.247	42.716	0.41%
	120	23.360	23.403	42.716	0.18%

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Ye, T.T. and Zhang, L.L. (2019 Derivatives Pricing via Machine Learning. *Journal of Mathematical Finance*, **9**, 561-589. https://doi.org/10.4236/jmf.2019.93029

[2] Han, J., Jentzen, A. and E, W.N. (2017) Overcoming the Curse of Dimensionality: Solving High-Dimensional Partial Differential Equations Using Deep Learning. Working Paper. https://arxiv.org/pdf/1707.02568.pdf

[3] Khoo, Y., Lu, J. and Ying, L. (2017) Solving Parametric PDE Problems with Artificial Neural Networks. Working Paper. https://arxiv.org/pdf/1707.03351.pdf

[4] Sirignano, J. and Spiliopoulos, K. (2017) DGM: A Deep Learning Algorithm for Solving Partial Differential Equations. Working Paper.

https://arxiv.org/pdf/1708.07469.pdf

[5] Beck, C., E, W.N. and Jentzen, A. (2017) Machine Learning Approximation Algorithms for High-Dimensional Fully Nonlinear Partial Deferential Equations and Second-Order Backward Stochastic Differential Equations. Working Paper. https://arxiv.org/pdf/1709.05963.pdf

[6] E, W.N., Han, J. and Jentzen, A. (2017) Deep Learning-Based Numerical Methods for High-Dimensional Parabolic Partial Deferential Equations and Backward Stochastic Differential Equations. Working Paper. https://arxiv.org/pdf/1706.04702.pdf

[7] Bouchard, B. and Elie, R. (2008) Discrete-Time Approximation of Decoupled Forward-Backward SDE with Jumps. *Stochastic Processes and Their Application*s, **118**, 53-75. https://doi.org/10.1016/j.spa.2007.03.010

[8] Fujii, M., Sato, S. and Takahashi, A. (2015) An FBSDE Approach to American Option Pricing with an Interacting Particle Method. *Asia-Pacific Financial Markets*, **22**, 239-260. https://doi.org/10.1093/rfs/14.1.113

[9] Longstaff, F.A. and Schwartz, E.S. (2001) Valuing American Options by Simulation: A Simple Least Square Approach. *Review of Financial Studies*, **14**,113-147. https://doi.org/10.1007/s10690-014-9195-6

Appendix: Proofs

Proof of Lemma 1. Denote $\lambda_i = \min_{\phi\in\Phi(\mathbb{D}_i)} \mathbb{E}\left[\left|\psi(X_T)-\phi(t,X_t)\right|^2 \mathbf{1}_{X_t\in\mathbb{D}_i}\right]$. Because we have $\mathbb{D}_i \subset \mathbb{D}_{i+1}$, it is obvious that $\Phi(\mathbb{D}_{i+1}) \subset \Phi(\mathbb{D}_i)$. Therefore, we have $\lambda_i < \lambda_{i+1} \le \lambda^*$, where

$$\lambda^* = \min_{\phi\in\Phi(\mathbb{D})} \mathbb{E}\left[\left|\psi(X_T)-\phi(t,X_t)\right|^2\right]. \tag{18}$$

We immediately have $\lim_{i\to\infty} \lambda_i = \hat{\lambda} \le \lambda^*$. In addition, we have

$$\lambda^* = \min_{\phi\in\Phi(\mathbb{D})} \mathbb{E}\left[\left|\psi(X_T)-\phi(t,X_t)\right|^2\right] \tag{19}$$

$$\le \min_{\phi_1\in\Phi(\mathbb{D}_i)} \mathbb{E}\left[\left|\psi(X_T)-\phi_1(t,X_t)\right|^2 \mathbf{1}_{X_t\in\mathbb{D}_i}\right] \tag{20}$$

$$+\mathbb{E}\left[\left|\psi(X_T)-\phi^*(t,X_t)\right|^2 \mathbf{1}_{X_t\in\mathbb{D}-\mathbb{D}_i}\right]. \tag{21}$$

Here $\phi^* = \arg\min_{\phi\in\Phi(\mathbb{D})} \mathbb{E}\left[\left|\psi(X_T)-\phi(t,X_t)\right|^2\right]$. The term

$$\mathbb{E}\left[\left|\psi(X_T)-\phi^*(t,X_t)\right|^2 \mathbf{1}_{X_t\in\mathbb{D}-\mathbb{D}_i}\right] \tag{22}$$

$$\le C\mathbb{P}\left(X_t \in \mathbb{D}-\mathbb{D}_i\right). \tag{23}$$

Here C is independent of i and we have $\lim_{i\to\infty} \mathbb{P}\left(X_t \in \mathbb{D}-\mathbb{D}_i\right)=0$. Combine this result with $\lambda_i < \lambda_{i+1} \le \lambda^*$ and we will obtain the claim of this lemma.

Lemma 2 (On Lead-Lag Regression). *Suppose that* $\mathcal{H} \subset \Phi$, *then we have*

$$\arg\min_{\phi\in\mathcal{H}} \mathbb{E}\left[\left|\psi(X_T)-\phi(t,X_t)\right|^2\right] = \arg\min_{\phi\in\mathcal{H}} \mathbb{E}\left[\left|\phi^*(t,X_t)-\phi(t,X_t)\right|^2\right] \tag{24}$$

where $\phi^*(t,X_t) = \arg\min_{\phi\in\Phi} \mathbb{E}\left[\left|\psi(X_T)-\phi(t,X_t)\right|^2\right]$.

Proof of Lemma 2. The proof of this lemma follows directly from the Repeated Projection Theorem (see, e.g., [1], Theorem 8).

Proof of Theorem 3. Apply Lemma 2 and ([1], Theorem 23) and the proof is obvious.

Proof of Theorem 4. The proof of the theorem follows from the error term of the Taylor expansion, Lemma 2 and ([1], Theorem 24).